

## Skiing

task: <b>skiing</b>	input file: <b>stdin</b>	output file: <b>stdout</b>
points: <b>100</b>	time limit: <b>1000 ms</b>	memory limit: <b>1 GB</b>

After an unsuccessful attempt to qualify for IOI, Kleofáš decided to become a slalom skiing champion. Tomorrow is a very important day in Kleofáš's life: he will compete in his very first skiing contest!

During the contest, Kleofáš will have to get from a starting point to a finishing point, while passing through  $n$  gates. In order to be as fast as possible, Kleofáš wants to use the shortest possible trajectory.

### Task

A skiing course can be described as a starting point  $S$ , a finishing point  $F$  and  $n$  gates. Each gate is a line segment parallel to the  $x$  axis (i. e. horizontal). No two gates are at the same  $y$  coordinate (altitude). The starting point is above each gate i. e. its  $y$  coordinate is higher than  $y$  coordinate of any gate. The finishing point is below each gate and below the starting point.

Find the shortest polygonal chain starting at point  $S$ , finishing at point  $F$  and intersecting all gates, in order **from top to bottom**. We say that a polygonal chain intersects a line segment if they have at least one common point (this point **can** be an endpoint of the line segment).

### Input

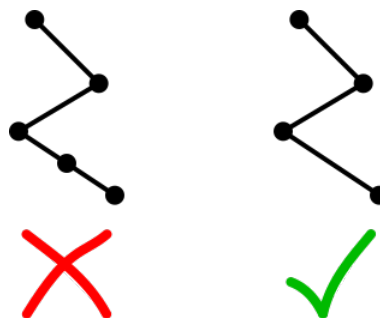
First line of the input contains one integer  $n$  ( $0 \leq n \leq 10^6$ ) – number of gates. Second line contains four integers  $x_S, y_S, x_F, y_F$ : coordinates of points  $S = (x_S, y_S)$  and  $F = (x_F, y_F)$  respectively.

$n$  lines follow,  $i$ -th of them contains three integers  $x_{1i}, x_{2i}, y_i$ , meaning that  $i$ -th gate is a segment from  $(x_{1i}, y_i)$  to  $(x_{2i}, y_i)$ . For each  $i$ ,  $x_{1i} < x_{2i}$  holds.

All coordinates are between  $-10^9$  and  $10^9$ , inclusive. Gates are ordered from top to bottom, i. e.  $y_S > y_1 > y_2 > \dots > y_n > y_F$ .

### Output

It can be proven that there always exists a unique shortest polygonal chain and all its vertices have integer coordinates. Output this chain without any redundant vertices (i. e. only output vertices where the chain changes its direction).



On the first line of the output print a single integer  $k$  – number of vertices in the optimal chain. Then print  $k$  more lines,  $i$ -th of them containing two space-separated integers  $x_i, y_i$  – coordinates of the  $i$ -th vertex of the chain. Vertices must be named from the start of the chain to the end, thus  $x_1 = x_S, y_1 = y_S, x_k = x_F, y_k = y_F$  and  $y_1 > y_2 > \dots > y_k$  must hold.

### Subtasks

subtask	points	maximum $n$
1	20	200
2	30	2000
3	50	1000000

Samples

input

```
4
5 10 6 0
0 4 7
7 10 6
5 8 4
2 5 1
```

output

```
5
5 10
4 7
7 6
5 1
6 0
```

The situation looks like this:

